IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): A.K. Iyengar et al.
Docket No.: YOR920040025US1
Serial No.: 10/804,516
Filing Date: March 19, 2004
Group: 2416
Examiner: Luat Phung

Title: Method and Apparatus for Dynamically Scheduling Requests

---

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants (hereinafter referred to as "Appellants") hereby appeal the final rejection dated March 16, 2010 (hereinafter "Office Action"), of claims 1-25 of the above-identified application.

REAL PARTY IN INTEREST

The assignee, International Business Machines Corporation, is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences.

STATUS OF CLAIMS

The present application was filed on March 19, 2004, with claims 1-25. Claims 1-25 remain pending, including independent claims 1, 14, 17, 18 and 25. Claims 1-25 stand finally rejected under 35 U.S.C. §103(a). Claims 1-25 are appealed.

STATUS OF AMENDMENTS

An amendment was filed on May 17, 2010, in which independent claims 1, 14 and 17 were amended. As indicated in the Advisory Action dated June 1, 2010, this amendment was entered.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method of processing a request to at least one server including a step of a processor receiving the request. The method also includes a step of the processor determining when to submit the request to the at least one server based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

In an illustrative embodiment described in the specification at, for example, page 6, lines 23-27; page 8, lines 13-15, with reference to FIG. 1; and page 16, line 6, to page 17, line 2, with reference to FIG. 3, a method of processing a request to at least one server (e.g., 180 in FIG. 1) includes a step of a processor (e.g., 100 in FIG. 1 and/or 310 in FIG. 3) receiving the request. As described in the specification at, for example, page 7, line 24, to page 8, line 1; page 9, lines 4-11, with reference to FIG. 1; and page 12, lines 8-21, with reference to FIG. 2, the method also includes a step of the processor determining when to submit the request to the at least one server. As described in the specification at, for example, page 7, lines 1-15; page 11, lines 1-15; and page 12, line 20, to page 13, line 22, this determination of when to submit the request to the at least one server is based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

Independent claim 14 is directed to an apparatus for processing a request to at least one server. The apparatus comprises a memory and at least one processor coupled to the memory. The at least one processor is operative to receive a request and to determine when to submit the request to the at least one server based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

In an illustrative embodiment described in the specification at, for example, page 6, lines 23-27; page 8, lines 13-15, with reference to FIG. 1; and page 16, line 6, to page 17, line 19, with reference to FIG. 3, an apparatus (e.g., 100 in FIG. 1 and/or 300 in FIG. 3) for processing a request to at least one server (e.g., 180 in FIG. 1) comprises a memory (e.g., 320 in FIG. 3) and at least one processor (e.g., 310 in FIG. 3) coupled to the memory, and the at least one processor is operative to receive a request. As described in the specification at, for example, page 7, line 24, to page 8, line 1; page 9, lines 4-11, with reference to FIG. 1; and page 12, lines 8-21, with reference to FIG. 2, the at least one processor is further operative to determine when to submit the request to the at least one server. As described in the specification at, for example, page 7, lines 1-15; page 11, lines 1-15; and page 12, line 20, to page 13, line 22, this determination of when to submit the request to the at least one server is based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

Independent claim 17 is directed to an article of manufacture for processing a request to at least one server comprising a computer readable medium containing one or more programs which when executed implement steps. These steps include receiving the request; and determining when to submit the request to the at least one server based on: (i) a quality-ofservice (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

In an illustrative embodiment described in the specification at, for example, page 6, lines 23-27; page 8, lines 13-15, with reference to FIG. 1; and page 17, lines 3-5 and 16-19, with reference to FIG. 3, an article of manufacture for processing a request to at least one server (e.g., 180 in FIG. 1)

comprises a computer readable medium (e.g., 320 in FIG. 3) containing one or more programs which when executed implement steps including receiving the request. As described in the specification at, for example, page 7, line 24, to page 8, line 1; page 9, lines 4-11, with reference to FIG. 1; and page 12, lines 8-21, with reference to FIG. 2, these steps also include determining when to submit the request to the at least one server. As described in the specification at, for example, page 7, lines 1-15; page 11, lines 1-15; and page 12, line 20, to page 13, line 22, this determination of when to submit the request to the at least one server is based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

Independent claim 18 is directed to a method of processing requests to at least one server. The method includes steps of assigning at least one client to a quality-of-service (QoS) class from among at least two QoS classes; assigning a response target to at least one QoS class; estimating at least one response time of the at least one server based on one or more requests sent to the server within a given time period; and a processor withholding submission of requests associated with a first one of the at least two QoS classes to allow requests associated with a second one of the at least two QoS classes to meet its response target based on the at least one estimated response time.

An illustrative embodiment includes a method of processing requests to at least one server (e.g., 180 in FIG. 1), as described in the specification at, for example, page 6, lines 23-27; page 8, lines 13-15, with reference to FIG. 1. As described in the specification at, for example, page 7, lines 1-17; page 8, lines 24-27, with reference to FIG. 1; the method includes a step of assigning at least one client to a quality-of-service (QoS) class from among at least two QoS classes. As described in the specification at, for example, page 7, lines 6-15; page 9, lines 1-3, with reference to FIG. 1, the method also includes a step of assigning a response target to at least one QoS class. As described in the specification at, for example, page 10, line 6, to page 11, line 21, the method additionally includes a step of estimating at least one response time of the at least one server based on one or more requests sent to the server within a given time period. As described in the specification at, for example, page 13, lines 8-17, with reference to FIG. 2, the method further includes a step of a processor (e.g., 100 in FIG. 1) withholding submission of requests associated with a first one of the

at least two QoS classes to allow requests associated with a second one of the at least two QoS classes to meet its response target based on the at least one estimated response time.

Independent claim 25 is directed to a method of providing a scheduling service for requests to at least one server. The method comprises the step of a service provider providing a scheduler comprising a processor. The processor is operative to: (i) assign at least one client to a quality-of-service (QoS) class from among at least two QoS classes; (ii) assign a response target to at least one QoS class; (iii) estimate at least one response time of the at least one server based on one or more requests sent to the server within a given time period; and (iv) withhold submission of requests associated with a first one of the at least two QoS classes to allow requests associated with a second one of the at least two QoS classes to meet its response target based on the at least one estimated response time.

In an illustrative embodiment described in the specification at, for example, page 6, lines 23-27; page 8, lines 13-15, with reference to FIG. 1; and page 15, line 24, to page 17, line 19, with reference to FIG. 3, a method of providing a scheduling service for requests to at least one server (e.g., 180 in FIG. 1) comprises the step of a service provider providing a scheduler (e.g., 100 in FIG. 1 and/or 300 in FIG. 3) comprising a processor (e.g., 310 in FIG. 3). As described in the specification at, for example, page 7, lines 1-17; page 8, lines 24-27, with reference to FIG. 1; the processor is operative to assign at least one client to a quality-of-service (QoS) class from among at least two QoS classes. As described in the specification at, for example, page 7, lines 6-15; page 9, lines 1-3, with reference to FIG. 1, the processor is also operative to assign a response target to at least one QoS class. As described in the specification at, for example, page 10, line 6, to page 11, line 21, the processor is further operative to estimate at least one response time of the at least one server based on one or more requests sent to the server within a given time period. As described in the specification at, for example, page 13, lines 8-17, with reference to FIG. 2, the processor is additionally operative to withhold submission of requests associated with a first one of the at least two QoS classes to allow requests associated with a second one of the at least two QoS classes to meet its response target based on the at least one estimated response time.

As described in the specification at, for example, page 3, lines 11-12, and page 13, lines 23-25, illustrative embodiments of the present invention advantageously offer efficient mechanisms for

achieving quality of service (QoS) goals such as specific response time targets, thereby improving throughput and system performance.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 5-9 and 14-17 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent Application Publication No. 2004/0162901 (hereinafter "Mangipudi") in view of U.S. Patent Application Publication No. 2005/0198200 (hereinafter "Subramanian").

2. Claims 2-4, 18-20 and 25 are rejected under 35 U.S.C. §103(a) as being unpatentable over Mangipudi and Subramanian in view of U.S. Patent No. 6,112,221 (hereinafter "Bender") and U.S. Patent Application Publication No. 2003/0120705 (hereinafter "Chen").

3. Claim 8 is rejected, in the alternative, under 35 U.S.C. §103(a) as being unpatentable over Mangipudi and Subramanian in view of U.S. Patent No. 6,807,156 (hereinafter "Veres").

4. Claims 10-12 are rejected under 35 U.S.C. §103(a) as being unpatentable over Mangipudi and Subramanian in view of U.S. Patent No. 6,981,029 (hereinafter "Menditto").

5. Claim 13 is rejected under 35 U.S.C. §103(a) as being unpatentable over Mangipudi, Subramanian and Menditto in view of U.S. Patent No. 6,772,211 (hereinafter "Lu").

6. Claims 21-23 are rejected under 35 U.S.C. §103(a) as being unpatentable over Mangipudi, Subramanian, Bender, Chen and Menditto.

7. Claim 24 is rejected under 35 U.S.C. §103(a) as being unpatentable over Mangipudi, Subramanian, Bender, Chen, Menditto and Lu.

ARGUMENT

1. Rejection of claims 1, 5-9 and 14-17 under §103 over Mangipudi and Subramanian.

Claims 1, 8 and 14-17

Claim 1 includes a limitation directed to a processor determining when to submit the request to the at least one server based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

In arguing that Mangipudi teaches determining when to submit a request to at least one server based on the factors recited in claim 1, the Examiner relies on Mangipudi at paragraph [0047], which describes "load sharing algorithms implemented in this illustrative embodiment to assign requests to back-end servers within a cluster. . . . All client requests sent to the illustrative embodiment are routed to the server selected as the most available and/or efficient server within each class according to a selected load balancing algorithm." It is important to note that the load balancing algorithm in paragraph [0047] of Mangipudi is being used solely to select <u>which</u> server client requests should be routed to, without determining <u>when</u> those requests should be submitted to the selected server.

Indeed, the Examiner concedes that paragraph [0047] of Mangipudi does not teach determining <u>when</u> those requests should be submitted to the at least one server. Rather, the Examiner argues that "scheduling a task refers to the timing of performing a task." Paragraph [0047] of Mangipudi does not describe an algorithm to be used for "scheduling," but rather describes "load sharing algorithms . . . to assign requests to back-end servers within a cluster."

The Examiner also describes paragraphs [0009] and [0011] of Mangipudi as disclosing "a well known technique of scheduling HTTP requests by placing them into queues [which] are serviced by the request controller based on configured policy such as length of queues, etc." This placement into queues affects in what <u>order</u> the requests will be serviced by the request controller but it does not determine <u>when</u> the requests should be submitted to a server in the manner recited in claim 1.

Even assuming *arguendo* that paragraph [0011] of Mangipudi could somehow be characterized as disclosing an arrangement which includes determining when to submit a request to a server, however, paragraph [0011] does not include any such determination which is based on the three factors recited in claim 1.

Moreover, paragraph [0011] of Mangipudi describes an <u>entirely different product</u> from that described in paragraph [0047]. Neither the first arrangement described in paragraph [0011] of Mangipudi nor the second arrangement described in paragraph [0047] of Mangipudi meet the limitations of claim 1 at issue. One skilled in the art would not have a reasonable expectation of success in combining these two entirely separate products without undue experimentation, nor would

one skilled in the art be motivated to attempt to combine these two products without improper hindsight.

Simply put, Mangipudi does not disclose any arrangement which includes determining when to submit the request to the at least one server based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server. Subramanian fails to remedy this fundamental deficiency of Mangipudi with regard to the limitations of claim 1.

Independent claims 14 and 17 recite limitations similar to those of claim 1 and are therefore believed to be similarly patentable.

Claim 8 is patentable at least by virtue of its dependency from independent claim 1, and claims 15 and 16 are patentable at least by virtue of their dependency from independent claim 14.

Claim 5

Claim 5 is patentable at least by virtue of its dependency from independent claim 1, the patentability of which has been heretofore discussed. Moreover, dependent claim 5 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 5 recites assigning the response target to the QoS class.

The Examiner argues that this limitation is taught by Mangipudi at paragraph [0054]. See the Office Action at page 7, last paragraph. The cited portion of Mangipudi states that "each back-end server in communication with the router is subject to a selectable one of a plurality of load balancing algorithms so that traffic is routed to the plurality of back-end servers, as a function of class, in a manner that maintains consistent response times and service level commitments even with increases in traffic and processing loads." In other words, Mangipudi is merely stating that traffic is routed as a function of class in a manner that maintains consistent response times despite increases in traffic processing loads. There is simply no teaching or suggestion directed to assigning a response target to any particular QoS class (i.e., a QoS class-specific response target) in the manner recited in claim 5.

Claim 6

Claim 6 is patentable at least by virtue of its dependency from claims 1 and 5, the patentability of which have been heretofore discussed. Moreover, dependent claim 6 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 6 specifies that the step of assigning the response target to the QoS class further comprises the step of assigning a response time target to the QoS class. An illustrative embodiment is described in the specification at, for example, page 7, lines 6-8.

The Examiner argues that this limitation is taught by Mangipudi at paragraph [0054]. See the Office Action at page 8, first paragraph. The cited portion of Mangipudi states that "each back-end server in communication with the router is subject to a selectable one of a plurality of load balancing algorithms so that traffic is routed to the plurality of back-end servers, as a function of class, in a manner that maintains consistent response times and service level commitments even with increases in traffic and processing loads." In other words, Mangipudi is merely stating that traffic is routed as a function of class in a manner that maintains consistent response times despite increases in traffic processing loads. There is simply no teaching or suggestion directed to assigning a response time target to any particular QoS class (i.e., a QoS class-specific response time target) in the manner recited in claim 6.

Claim 7

Claim 7 is patentable at least by virtue of its dependency from claims 1 and 5, the patentability of which have been heretofore discussed. Moreover, dependent claim 7 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 7 specifies that the step of assigning the response target to the QoS class further comprises the step of assigning a response percentile target to the QoS class. An illustrative embodiment is described in the specification at, for example, page 7, lines 13-15.

The Examiner argues that this limitation is taught by Mangipudi at paragraph [0048]. See the Office Action at page 8, second paragraph. The cited portion of Mangipudi states that the "weighted percentage load balancing predictor option allows the site administrator to assign a performance

weight to each server. Weighted load balancing is similar to least connections, but servers with a higher weight value will receive a larger percentage of connections at any one time." In other words, Mangipudi is merely stating that a server with a higher weight value will receive a larger percentage of connections at any one time. There is simply no teaching or suggestion directed to assigning a response percentile target to any particular QoS class (i.e., a QoS class-specific response percentile target) in the manner recited in claim 7.

Claim 9

Claim 9 is patentable at least by virtue of its dependency from independent claim 1, the patentability of which has been heretofore discussed. Moreover, dependent claim 9 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 9 recites assigning a target response time to a plurality of QoS classes in which lower quality classes are assigned larger response times than higher quality classes.

The Examiner argues that this limitation is taught by Mangipudi at paragraph [0038]. See the Office Action at page 8, fourth paragraph. The cited portion of Mangipudi states that "assigning more resources to a cluster to support higher class requests guarantees that more resources are available to this class, and this class is given priority as compared to other classes." However, there is simply no teaching or suggestion directed to assigning a target response time to any particular QoS class (i.e., a QoS class-specific target response time) in the manner recited in claim 9.

2. Rejection of claims 2-4, 18-20 and 25 under §103 over Mangipudi, Subramanian, Bender and Chen.

Claim 2

Claim 2 includes a limitation directed to withholding submission of a request when the request originated from a client assigned to a first QoS class to allow a request that originated from a client assigned to a second QoS class to meet a response target associated therewith. An illustrative embodiment described in the specification at, for example, page 13, lines 8-17:

> [C]onsider a workload including not only type 2 requests [(response time targets)], but also type 3 requests (best effort). Recall that in this case, the goal is to achieve good performance for the type 3 requests without violating the response time targets of the type 2 requests. The scheduler 140 does so by maintaining two queues. One that keeps type 2 requests sorted by their dispatch deadlines and one that keeps the type 3 requests sorted in First-Come-First-Serve (FCFS) order. When choosing a request for dispatching, the scheduler first compares the current time to the dispatch deadline of the first request in the type 2 queue. If the dispatch deadline is met or already exceeded, the first request in the type 2 queue is dispatched. Otherwise, the first request from the type 3 queue is dispatched.

Appellants note that the Examiner concedes that Mangipudi and Subramanian fail to disclose the aforementioned limitation of claim 2. Indeed, as discussed above, Mangipudi only discloses determining which server a request should be submitted to, not when a request should be submitted to that server, and thus fails to teach or request any withholding submission of a request.

Rather, the Examiner argues that Bender discloses the limitation at issue. It is important to note that Bender deals exclusively with scheduling execution of jobs which either have already been submitted to the server (which Bender refers to as "on-line" scheduling) or which will be submitted at a definite future time (which Bender refers to as "off-line" scheduling). See Bender at column 3, lines 23-35. Bender does not teach or suggest any technique which involves withholding submission of requests to the server. Chen similarly does not involve withholding submission of requests to a server. Moreover, as explained in Bender at column 5, lines 27-43:

> At step 108, once the deadline for each uncompleted job is calculated, server system 10 schedules the jobs in accordance with an earliest deadline first ("EDF") methodology. With an EDF methodology, the first job that server system 10 schedules is the job which has the earliest deadline, as found in step 106, relative to all of the other jobs. It then chooses the job with the next earliest deadline, and schedules it second, and so on until all of the jobs have been scheduled.
>
> At decision step 110, server system 10 inquires whether each and every one of the jobs have completion times which is earlier than each job's respective deadline, as found in step 106. If any job is not able to be completed prior to its deadline, then the estimated stretch value is not feasible and is therefore adjusted at step 112. From step 112, the feasibility of the adjusted stretch value is re-checked by returning to step 106.

Thus, Bender schedules a job based on completion times and deadline times associated with *that particular job*. Bender does not withhold the request from submission to the at least one server when the request originated from a client assigned to a first QoS class to allow a request that originated from a client assigned to a second QoS class to meet a response target associated therewith. That is, Bender does not withhold a job based on a "response target" associated with *a particular QoS class*. Chen fails to remedy this fundamental deficiency of Mangipudi, Subramanian and Bender. Thus, Bender and Chen fail to remedy the admitted failure of Mangipudi to reach the limitations of claim 2.

<u>Claim 3 and 19</u>

Claim 3 is patentable at least by virtue of its dependency from claims 1 and 2, the patentability of which have been heretofore discussed. Moreover, dependent claim 3 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 3 includes a limitation directed to <u>reducing a request withhold rate to increase throughput</u> of the at least one server. An illustrative embodiment is described in the specification at, for example, page 11, line 22, to page 12, line 2:

> In addition to the observed mean response times, a second equally important consideration in adjusting the MPL is the system throughput. Setting the MPL to too small values might reduce the utilization of system resources at the server to a degree that results in loss of system throughput. Thus, the MPL adaptor 150 monitors, in addition to response times, the system throughput based on the request arrival rate in the monitoring interval. If after reducing the MPL target, the system throughput drops, even though the arrival rate stayed the same or increased compared to previous monitoring intervals, the MPL adaptor 150 increases the current MPL target again.

The Examiner concedes that Mangipudi, and presumably Subramanian and Chen, fails to disclose the aforementioned limitation of claim 3. See the Office Action at page 11, last paragraph, and page 12, first paragraph. Rather, the Examiner relies on Bender's alleged disclosure of "a server which employs a pre-emptive setting of scheduling requests according to an earliest deadline first

methodology, calculates processing time and dead line for each request, and continues adjusting estimated processing time." Office Action at page 12, third paragraph (citations omitted).

Even assuming that the Examiner has accurately characterized Bender's disclosure in the above-quoted portion of the Office Action, however, Bender is silent as to <u>reducing a request withhold rate</u>, much less doing so <u>to increase throughput</u>. Appellants note that Bender's pre-emptive earliest deadline first (EDF) scheduling deals only with the order in which a plurality of different pending jobs are to be processed, not with whether to withhold a request.

See Bender at column 4, lines 52-58: "server system 10 employs a pre-emptive setting. In such a setting, the server does not service a particular job request continuously until the particular job request is complete. Instead, server system 10 alternates between a plurality of different pending jobs, servicing portions of each job until eventually all of the jobs are complete."

See also Bender at column 5, lines 27-35: "At step 108, once the deadline for each uncompleted job is calculated, server system 10 schedules the jobs in accordance with an earliest deadline first ('EDF') methodology. With an EDF methodology, the first job that server system 10 schedules is the job which has the earliest deadline, as found in step 106, relative to all of the other jobs. It then chooses the job with the next earliest deadline, and schedules it second, and so on until all of the jobs have been scheduled."

Thus, claim 3 is believed to be separately patentable over the cited references.

<u>Claim 4</u>

Claim 4 is patentable at least by virtue of its dependency from claims 1 and 2, the patentability of which have been heretofore discussed. Moreover, dependent claim 4 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 4 includes a limitation directed to <u>varying a request withhold rate to balance the throughput and request response times</u>. In an illustrative embodiment described in the specification at, for example, page 7, lines 1-4, "the scheduler may control (e.g., varies by increasing or decreasing) a request withhold rate (e.g., a rate of requests withheld from submission to a server) and thus the number of requests sent to a server so as to, for example, balance server throughput and request response times."

The Examiner concedes that Mangipudi, and presumably Subramanian and Chen, fails to disclose the aforementioned limitation of claim 3. See the Office Action at page 11, last paragraph, and page 12, first paragraph. Rather, the Examiner relies on Bender's alleged disclosure of "a server which employs a pre-emptive setting of scheduling requests according to an earliest deadline first methodology, calculates processing time and dead line for each request, and continues adjusting estimated processing time." Office Action at page 12, third paragraph (citations omitted).

Even assuming that the Examiner has accurately characterized Bender's disclosure in the above-quoted portion of the Office Action, however, Bender is silent as to <u>varying a request withhold rate</u>, much less doing so <u>to balance the throughput and request response times</u>. Appellants note that Bender's pre-emptive earliest deadline first (EDF) scheduling deals only with the order in which a plurality of different pending jobs are to be processed, not with whether to withhold a request.

See Bender at column 4, lines 52-58: "server system 10 employs a pre-emptive setting. In such a setting, the server does not service a particular job request continuously until the particular job request is complete. Instead, server system 10 alternates between a plurality of different pending jobs, servicing portions of each job until eventually all of the jobs are complete."

See also Bender at column 5, lines 27-35: "At step 108, once the deadline for each uncompleted job is calculated, server system 10 schedules the jobs in accordance with an earliest deadline first ('EDF') methodology. With an EDF methodology, the first job that server system 10 schedules is the job which has the earliest deadline, as found in step 106, relative to all of the other jobs. It then chooses the job with the next earliest deadline, and schedules it second, and so on until all of the jobs have been scheduled."

Thus, claim 4 is believed to be separately patentable over the cited references.

<u>Claims 18 and 25</u>

Independent claims 18 and 25 include both limitations similar to those discussed above with reference to independent claim 1 and limitations similar to those discussed above with reference to dependent claim 2. As such, independent claims 18 and 25 are believed to be patentable at least for reasons similar to those discussed above with reference to claims 1 and 2.

Claim 19

Claim 19 is patentable at least by virtue of its dependency from claim 18, the patentability of which has been heretofore discussed. Moreover, dependent claim 19 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 19 includes a limitation similar to that discussed above with reference to claim 3, and hence is similarly patentable.

Claim 20

Claim 20 is patentable at least by virtue of its dependency from claim 18, the patentability of which has been heretofore discussed. Moreover, dependent claim 20 is believed to recite separately patentable subject matter over the cited references. Specifically, claim 20 includes a limitation similar to that discussed above with reference to claim 4, and hence is similarly patentable.

3. Rejection of claim 8 under §103 over Mangipudi, Subramanian and Veres.

Claim 8 is patentable at least by virtue of its dependency from claim 1, the patentability of which has been heretofore discussed. Moreover, Veres fails to remedy the aforementioned deficiencies of Mangipudi and Subramanian with regard to the limitations of claim 1.

4. Rejection of claims 10-12 under §103 over Mangipudi, Subramanian and Menditto.

Claims 10-12 are patentable at least by virtue of their dependency from claim 1, the patentability of which has been heretofore discussed. Moreover, Menditto fails to remedy the aforementioned deficiencies of Mangipudi and Subramanian with regard to the limitations of claim 1.

5. Rejection of claim 13 under §103 over Mangipudi, Subramanian, Menditto and Lu.

Claim 13 are patentable at least by virtue of its dependency from claim 1, the patentability of which has been heretofore discussed. Moreover, Menditto and Lu fail to remedy the aforementioned deficiencies of Mangipudi and Subramanian with regard to the limitations of claim 1.

6. Rejection of claims 21-23 under §103 over Mangipudi, Subramanian, Bender, Chen and Menditto.

Claims 21-23 are patentable at least by virtue of their dependency from claim 18, the patentability of which has been heretofore discussed. Moreover, Menditto fails to remedy the aforementioned deficiencies of Mangipudi, Subramanian, Bender and Chen with regard to the limitations of claim 18.

7. Rejection of claim 24 under §103 over Mangipudi, Subramanian, Bender, Chen, Menditto and Lu.

Claim 24 is patentable at least by virtue of its dependency from claim 18, the patentability of which has been heretofore discussed. Moreover, Menditto and Lu fail to remedy the aforementioned deficiencies of Mangipudi, Subramanian, Bender and Chen with regard to the limitations of claim 18.

In view of the above, Appellants believe that claims 1-25 are in condition for allowance, and respectfully request reversal of the §103(a) rejections.

Respectfully submitted,

/des/

Date: September 16, 2010

David E. Shifren
Attorney for Applicant(s)
Reg. No. 59,329
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2641

CLAIMS APPENDIX

1. A method of processing a request to at least one server, comprising the steps of:

a processor receiving the request; and

the processor determining when to submit the request to the at least one server based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

2. The method of claim 1, further comprising the step of withholding the request from submission to the at least one server when the request originated from a client assigned to a first QoS class to allow a request that originated from a client assigned to a second QoS class to meet a response target associated therewith.

3. The method of claim 2, further comprising the steps of:

determining a throughput of the at least one server; and

reducing a request withhold rate to increase throughput of the at least one server.

4. The method of claim 2, further comprising the steps of:

monitoring a throughput of the at least one server; and

varying a request withhold rate to balance the throughput and request response times.

5. The method of claim 1, further comprising the step of assigning the response target to the QoS class.

6. The method of claim 5, wherein the step of assigning the response target to the QoS class further comprises the step of assigning a response time target to the QoS class.

7. The method of claim 5, wherein the step of assigning the response target to the QoS class further comprises the step of assigning a response percentile target to the QoS class.

8. The method of claim 1, further comprising the step of estimating the response time associated with the at least one server based on one or more requests sent to the at least one server within a given time period.

9. The method of claim 1, further comprising the step of assigning a target response time to a plurality of QoS classes in which lower quality classes are assigned larger response times than higher quality classes.

10. The method of claim 1, further comprising the steps of:

determining dispatch times for requests from a difference between at least one predicted response time of the at least one server and the target response time corresponding to the QoS class of the request; and

sending requests to the at least one server based on dispatch times.

11. The method of claim 1, wherein a plurality of applications are running on the at least one server and requests are routed to applications, further comprising the steps of:

estimating response times of applications based on one or more requests sent to the applications within a time period; and

sending a request to an application whose estimated response time is not greater than a target response time corresponding to the QoS class of the request.

12. The method of claim 11, further comprising the step of varying a number of requests sent to applications so that estimated response times of applications are not greater than target response times of QoS classes corresponding to requests sent to the applications.

13. The method of claim 11, wherein the at least one server comprises a plurality of servers and each application runs on a different one of the plurality of servers.

14. Apparatus for processing a request to at least one server, comprising:

a memory; and

at least one processor coupled to the memory and operative to receive a request, and determine when to submit the request to the at least one server based on: (i) a quality-of-service

(QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

15. The apparatus of claim 14, wherein the memory and the at least one processor form a scheduler that is external to the at least one server.

16. The apparatus of claim 15, wherein the scheduler is a front-end scheduler and the at least one server is a back-end server.

17. An article of manufacture for processing a request to at least one server, comprising a computer readable medium containing one or more programs which when executed implement the steps of:

receiving the request; and

determining when to submit the request to the at least one server based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

18. A method of processing requests to at least one server, comprising the steps of:

assigning at least one client to a quality-of-service (QoS) class from among at least two QoS classes;

assigning a response target to at least one QoS class;

estimating at least one response time of the at least one server based on one or more requests sent to the server within a given time period; and

a processor withholding submission of requests associated with a first one of the at least two QoS classes to allow requests associated with a second one of the at least two QoS classes to meet its response target based on the at least one estimated response time.

19. The method of claim 18, further comprising the steps of:

determining a throughput of the at least one server; and

reducing a request withhold rate to increase throughput of the at least one server.

20. The method of claim 18, further comprising the steps of:

monitoring a throughput of the at least one server; and

varying a request withhold rate to balance the throughput and request response times.

21. The method of claim 18, further comprising the steps of:

determining dispatch times for requests from a difference between at least one predicted response time of the at least one server and the target response time corresponding to the QoS class of the request; and

sending requests to the at least one server based on dispatch times.

22. The method of claim 18, wherein a plurality of applications are running on the at least one server and requests are routed to applications, further comprising the steps of:

estimating response times of applications based on one or more requests sent to the applications within a time period; and

sending a request to an application whose estimated response time is not greater than a target response time corresponding to the QoS class of the request.

23. The method of claim 22, further comprising the step of varying a number of requests sent to applications so that estimated response times of applications are not greater than target response times of QoS classes corresponding to requests sent to the applications.

24. The method of claim 22, wherein the at least one server comprises a plurality of servers and each application runs on a different one of the plurality of servers.

25. A method of providing a scheduling service for requests to at least one server, comprising the step of:

a service provider providing a scheduler comprising a processor operative to: (i) assign at least one client to a quality-of-service (QoS) class from among at least two QoS classes; (ii) assign a response target to at least one QoS class; (iii) estimate at least one response time of the at least one server based on one or more requests sent to the server within a given time period; and (iv) withhold submission of requests associated with a first one of the at least two QoS classes to allow

requests associated with a second one of the at least two QoS classes to meet its response target based on the at least one estimated response time.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.